```
// This source code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/M
// © TheTrdFloor
//
// In many strategies, it's quite common to use a scaled ATR to help define a stop-loss, and it's not
// as well.  While there are quite a few indicators that plot ATR bands already on TV, we could not fi
// way that we wanted.  They all had at least one of the following gaps:
//   * The ATR offset was not configurable (usually hard-coded to be based off the high or low, while
//   * It would only print a single band (either the upper or lower), which would require the same ind
//   * The ATR scaling factor was either not configurable or only stepped in whole numbers (often time
//
// Also, when looking at some of the behaviors of the ATR bands, you can see that when price first lev
// the first peak of the upper ATR band to the first valley of the lower ATR band and look for price t
// happens, price will usually make a notable move in that direction.
//
// While we have made some updates and enhancements to this indicator, and have every intention of con
// for enhancement, credit is still due to the origianl author:  AlexanderTeaH
//
//@version=5
indicator('ATR Bands', overlay=true) // These had to be taken out as they're not compatible with plott

// Inputs
atrPeriod = input.int(title='ATR Period', defval=3, minval=1, group="ATR Bands Standard Settings", too
                "Most often this is set at either 14 or 21.\nDefault: 3")

// While it seemed like a nice idea at the time, having separately-configurable upper and lower bands
// Therefore, We're going to simplify the config to make these settings unified for both bands, as it
//
// atrMultiplierUpper = input.float(title='ATR Upper Band Scale Factor', defval=2.5, step=0.1, minval=
//                     "This will usually be between 1 and 3.")
// srcUpper = input.source(title='ATR Upper Offset Source', defval=close, group="ATR Upper Band Settin
//                     "For this band, 'high' and 'close' (default) are generally the most appropriate
//
atrMultiplier = input.float(title='ATR Band Scale Factor', defval=2.5, step=0.1, minval=0.01, group="A
                    "This will usually be between 1 and 3.\n\nDefault: 2.5")
// On second thought, I'm going to nix this setting and force it to be the "close" source.  Having the
atrSourceRef = "close"
//atrSourceRef = input.string(title='ATR Upper Offset Source', defval="close", options=["close","wicks
//                     "The default value 'close' should be your go-to, but 'wicks' might provide a bi
//
// See above - these are deprecated and no longer used...
//
// atrMultiplierLower = input.float(title='ATR Lower Band Scale Factor', defval=2.5, step=0.1, minval=
//                     "This will usually be between 1 and 3.")
// srcLower = input.source(title='ATR Lower Offset Source', defval=close, group="ATR Lower Band Settin
//                     "For this band, 'low' and 'close' (default) are generally the most appropriate
//
//
// Take-Profit band settings
showTPBands = input.bool(title="Show opposite bands for take-profit zones", defval=false, tooltip="If
                    "to depict potential take-profit targets that are scaled based on the 'stop-loss'
tpScaleFactor = input.float(title="Take-Profit Scale Factor", defval=1.5, minval=1, step=0.1, tooltip=
                    "The easiest way to think of this is as a desired reward/risk ratio, where the pr
```

```
//
//
// As an added bonus, give the option to plot a table containing exact figures for all of the bands...
// Functional settings
showTable = input.bool(title="Show Table for Stops and Targets", defval=false, group="Table Settings",
                    "Note: Take-profit values are based on the 'Take-Profit Scale Factor' setting abc
allowTableRepainting = input.bool(title="Allow Table Repainting", defval=false, group="Table Settings"
                    "to candle close, but should be used with extreme caution.\n\nDefault: Unchecked"
showTPinTable = input.bool(title="Include additional rows/columns to display take-profit values.", def
                    "Note: Take-profit values are based on the 'Take-Profit Scale Factor' setting abc
//
// Display settings
alignTableVertically = input.bool(title="Align Table Vertically", defval=true, group="Table Settings",
tablePosition = input.string(title="Table Location", defval="Bottom Right", options=["Top Right","Top
                    tooltip='This setting controls the position on the chart where the table will be
tableColor = input.color(title="Table Color: ", defval=color.rgb(0, 175, 200, 20), group="Table Settir
tableTextHeaderColor = input.color(title="Table Header Color: ", defval=color.rgb(255, 255, 0, 0), gro
tableTextColor = input.color(title="Table Text Color: ", defval=color.rgb(255, 255, 255, 0), group="Ta
// tableTooltipColor = input.color(title="Table Tooltip Color: ", defval=color.rgb(255, 75, 255, 0), g
tableLongBGColor = input.color(title="Table Background Color - Long: ", defval=color.rgb(0, 255, 0, 90
tableShortBGColor = input.color(title="Short: ", defval=color.rgb(255, 0, 0, 80), group="Table Setting


// Functions
//
// Function to convert the input "source" to a proper "source"
getBandOffsetSource(srcIn, isUpperBand) =>
    // Initialize the return to our fail-safe 'close', which is also the default input, then update th
    ret = close
    switch srcIn
        "close" => ret := close
        "wicks" => ret := isUpperBand ? high : low
        => ret := close
    ret
//
// Function to convert table position input to a an appropriate argument
getTablePosition(posIn) =>
    posOut = position.bottom_right
    switch (posIn)
        "Top Right" => posOut := position.top_right
        "Top Left" => posOut := position.top_left
        "Top Center" => posOut := position.top_center
        "Middle Right" => posOut := position.middle_right
        "Middle Left" => posOut := position.middle_left
        "Middle Center" => posOut := position.middle_center
        "Bottom Right" => posOut := position.bottom_right
        "Bottom Left" => posOut := position.bottom_left
        "Bottom Center" => posOut := position.bottom_center
        => posOut := position.bottom_right
    posOut

// ATR
atr = ta.atr(atrPeriod)
scaledATR = atr * atrMultiplier
upperATRBand = getBandOffsetSource(atrSourceRef, true) + scaledATR
```

```
lowerATRBand = getBandOffsetSource(atrSourceRef, false) - scaledATR
//
// Since we can calcualte ATR bands based on either close or wicks, we need to be sure to normalize th
// from the close to the "stop band" before we can then apply our take-profit scaler and calculate the
scaledTPLong = close + ((close - lowerATRBand) * tpScaleFactor)
scaledTPShort = close - ((upperATRBand - close) * tpScaleFactor)

// OG ATR Band Plotting
plot(upperATRBand, title="Upper ATR Band", color=color.rgb(0, 255, 0, 50), linewidth=2)
plot(lowerATRBand, title="Lower ATR Band", color=color.rgb(255, 0, 0, 50), linewidth=2)

// TP band plots
plot(showTPBands ? scaledTPLong : na, title="Upper Take-Profit Band", color=color.rgb(255, 255, 255, 8
plot(showTPBands ? scaledTPShort : na, title="Lower Take-Profit Band", color=color.rgb(255, 255, 0, 80

// ATR and TP table...
if (showTable)
    // It's nice that TV will automagically shrink/reposition table cells to not have gaps if a specif
    // so we can define the table to the max number of rows/columns possible for this indicator in any
    var atrTable = table.new(position=getTablePosition(tablePosition), columns=8, rows=8)
    //
    // Set the base table styles...
    table.set_border_width(atrTable, 1)
    table.set_frame_width(atrTable, 1)
    table.set_border_color(atrTable, tableColor)
    table.set_frame_color(atrTable, tableColor)
    //
    // Since we're giving the option to display the table with 2 different formats (horizontal vs vert
    // incorporate a method to switch from one to the other based on the 'alignTableVertically' user i
    // conditional logic inside the 'table.cell' functions, it will be far more intuitive to "read" if
    //
    // While this WILL result in a pretty notable duplication of code, it's acceptable in this case as
    //
    // Vertical orientation
    if (alignTableVertically)
        // Define the Title/Header cells
        table.cell(atrTable, 0, 0, text="Long ATR Stop", text_color=tableTextHeaderColor, bgcolor=tabl
        table.cell(atrTable, 0, 1, text="Long ATR Stop Dist", text_color=tableTextHeaderColor, bgcolor
        if (showTPinTable)
            table.cell(atrTable, 0, 2, text="Long ATR TP", text_color=tableTextHeaderColor, bgcolor=ta
            // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
            if (tpScaleFactor != 1)
                table.cell(atrTable, 0, 3, text="Long ATR TP Dist", text_color=tableTextHeaderColor, b
        table.cell(atrTable, 0, 4, text="Short ATR Stop", text_color=tableTextHeaderColor, bgcolor=tab
        table.cell(atrTable, 0, 5, text="Short ATR Stop Dist", text_color=tableTextHeaderColor, bgcolo
        if (showTPinTable)
            table.cell(atrTable, 0, 6, text="Short ATR TP", text_color=tableTextHeaderColor, bgcolor=t
            // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
            if (tpScaleFactor != 1)
                table.cell(atrTable, 0, 7, text="Short ATR TP Dist", text_color=tableTextHeaderColor,
        //
        // Now for table values for each header...
        // Start with Long position...
        table.cell(atrTable, 1, 0, text=str.tostring(allowTableRepainting ? lowerATRBand : lowerATRBar
        table.cell(atrTable, 1, 1, text=str.tostring(math.round_to_mintick(allowTableRepainting ? clos
```

```
    if (showTPinTable)
        table.cell(atrTable, 1, 2, text=str.tostring(allowTableRepainting ? scaledTPLong : scaledT
        // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
        if (tpScaleFactor != 1)
            table.cell(atrTable, 1, 3, text=str.tostring(math.round_to_mintick(allowTableRepaintir
    // Now the Short position...
    table.cell(atrTable, 1, 4, text=str.tostring(allowTableRepainting ? upperATRBand : upperATRBar
    table.cell(atrTable, 1, 5, text=str.tostring(math.round_to_mintick(allowTableRepainting ? uppe
    if (showTPinTable)
        table.cell(atrTable, 1, 6, text=str.tostring(allowTableRepainting ? scaledTPShort : scaled
        // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
        if (tpScaleFactor != 1)
            table.cell(atrTable, 1, 7, text=str.tostring(math.round_to_mintick(allowTableRepaintir



//
// Horizontal orientation
else
    // Define the Title/Header cells
    table.cell(atrTable, 0, 0, text="Long ATR Stop", text_color=tableTextHeaderColor, bgcolor=tabl
    table.cell(atrTable, 1, 0, text="Long ATR Stop Dist", text_color=tableTextHeaderColor, bgcolor
    if (showTPinTable)
        table.cell(atrTable, 2, 0, text="Long ATR TP", text_color=tableTextHeaderColor, bgcolor=ta
        // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
        if (tpScaleFactor != 1)
            table.cell(atrTable, 3, 0, text="Long ATR TP Dist", text_color=tableTextHeaderColor, b
    table.cell(atrTable, 4, 0, text="Short ATR Stop", text_color=tableTextHeaderColor, bgcolor=tab
    table.cell(atrTable, 5, 0, text="Short ATR Stop Dist", text_color=tableTextHeaderColor, bgcolo
    if (showTPinTable)
        table.cell(atrTable, 6, 0, text="Short ATR TP", text_color=tableTextHeaderColor, bgcolor=t
        // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
        if (tpScaleFactor != 1)
            table.cell(atrTable, 7, 0, text="Short ATR TP Dist", text_color=tableTextHeaderColor,
    //
    // Now for table values for each header...
    // Start with Long position...
    table.cell(atrTable, 0, 1, text=str.tostring(allowTableRepainting ? lowerATRBand : lowerATRBar
    table.cell(atrTable, 1, 1, text=str.tostring(math.round_to_mintick(allowTableRepainting ? clos
    if (showTPinTable)
        table.cell(atrTable, 2, 1, text=str.tostring(allowTableRepainting ? scaledTPLong : scaledT
        // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
        if (tpScaleFactor != 1)
            table.cell(atrTable, 3, 1, text=str.tostring(math.round_to_mintick(allowTableRepaintir
    // Now the Short position...
    table.cell(atrTable, 4, 1, text=str.tostring(allowTableRepainting ? upperATRBand : upperATRBar
    table.cell(atrTable, 5, 1, text=str.tostring(math.round_to_mintick(allowTableRepainting ? uppe
    if (showTPinTable)
        table.cell(atrTable, 6, 1, text=str.tostring(allowTableRepainting ? scaledTPShort : scaled
        // If the TP scale factor is exactly 1, we can nix the TP distance columns as it will be e
        if (tpScaleFactor != 1)
            table.cell(atrTable, 7, 1, text=str.tostring(math.round_to_mintick(allowTableRepaintir
```

---

*PDF* document made with CodePrint using Prism